

Experiment Report 2: Hardware Comparison on Tiny Shakespeare (Character-Level nanoGPT)

Experimental Setup

Date: Thursday, 02/05/2026

Participants: John Lee, Maximo Sanchez

In this experiment, we trained a **tiny character-level GPT model** on the *Tiny Shakespeare* dataset to evaluate whether differences in hardware specifications (CPU vs GPU, and across GPU tiers) lead to observable differences in training behavior at **medium** model scales. Bumping up the heads, layers, and embedding values for extra complexity

All participants trained the same model using the **nanoGPT framework** within the shared COMP560-jmac repository. To ensure experimental fairness and reproducibility, all runs were logged to a shared Weights & Biases project: **GPUvsCPUResearch** (<https://wandb.ai/dickinson-comp560-sp26/GPUvsCPUResearch>)

The experiment was conducted on two machines with differing hardware capabilities:

- **RTX 3070 GPU, AMD Ryzen 7 5800H system**
- **RTX 4060 GPU, AMD Ryzen 7 8845HS system,**

These systems were ordered conceptually from highest to lowest theoretical compute capability. All models were trained for the same number of iterations using identical configurations, seeds, and dataset splits.

Model Configuration (Held Constant Across All Runs)

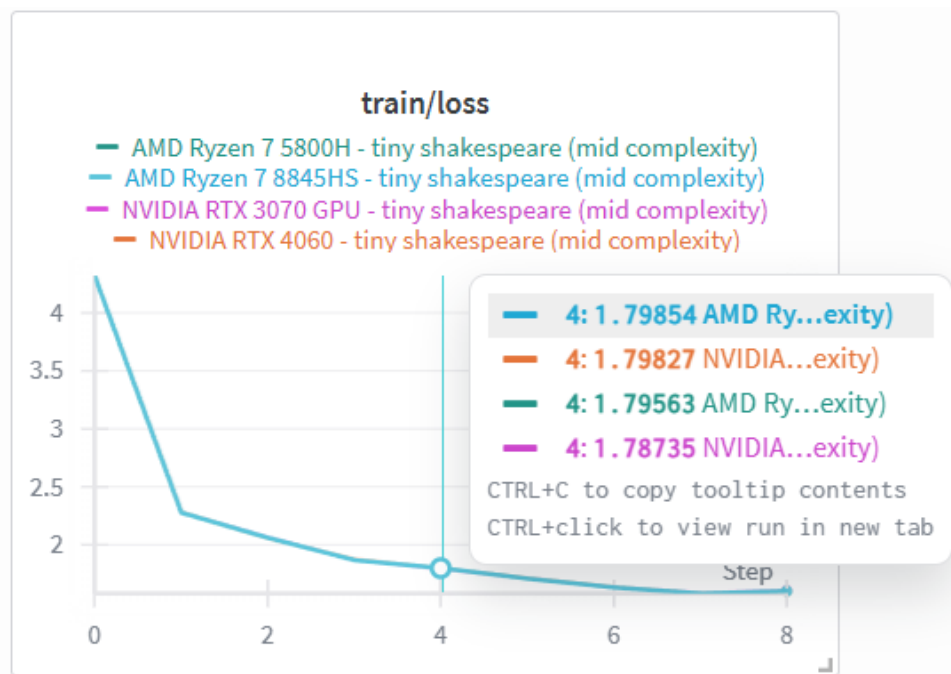
To isolate the effect of hardware, **all hyperparameters were fixed** across devices:

```
# embryonic GPT model
n_layer = 6
n_head = 6
n_embd = 384 # need n_embd % n_head == 0
dropout = 0.0
learning_rate = 1e-3 # with baby networks can afford to go a bit higher
max_iters = 2000
lr_decay_iters = 2000 # make equal to max_iters usually
min_lr = 1e-4 # learning_rate / 10 usually
beta2 = 0.99 # make a bit bigger because number of tokens per iter is small
warmup_iters = 100 # not super necessary potentially
```

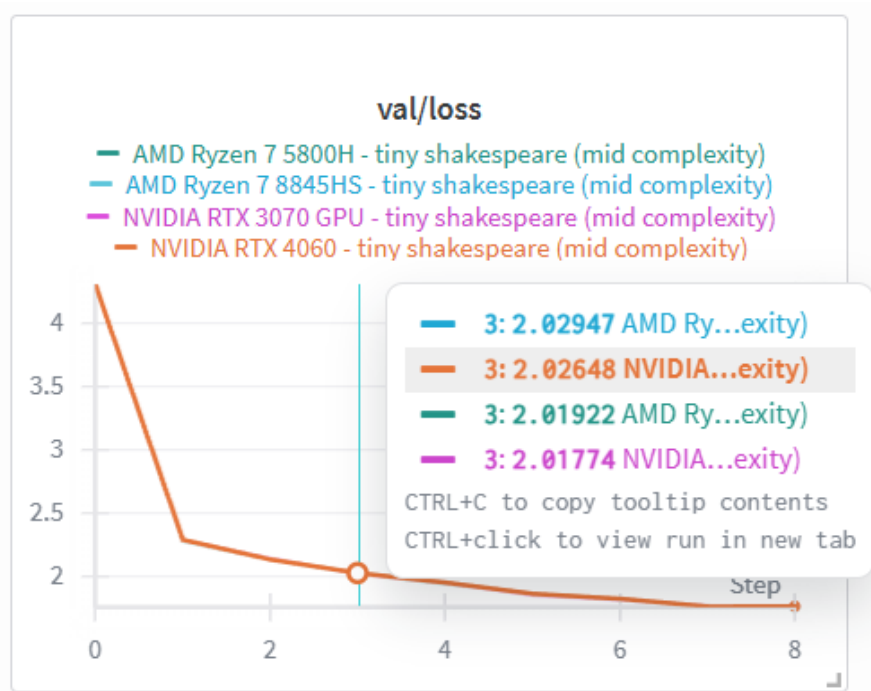
We have scaled up to a **Medium model (6 layers, 6 heads, 384 embd)**. This configuration represents a '**Transition Zone**'—a mid-sized bridge designed to identify the exact crossover point where GPU efficiency begins to overtake the CPU.

Results and Observations

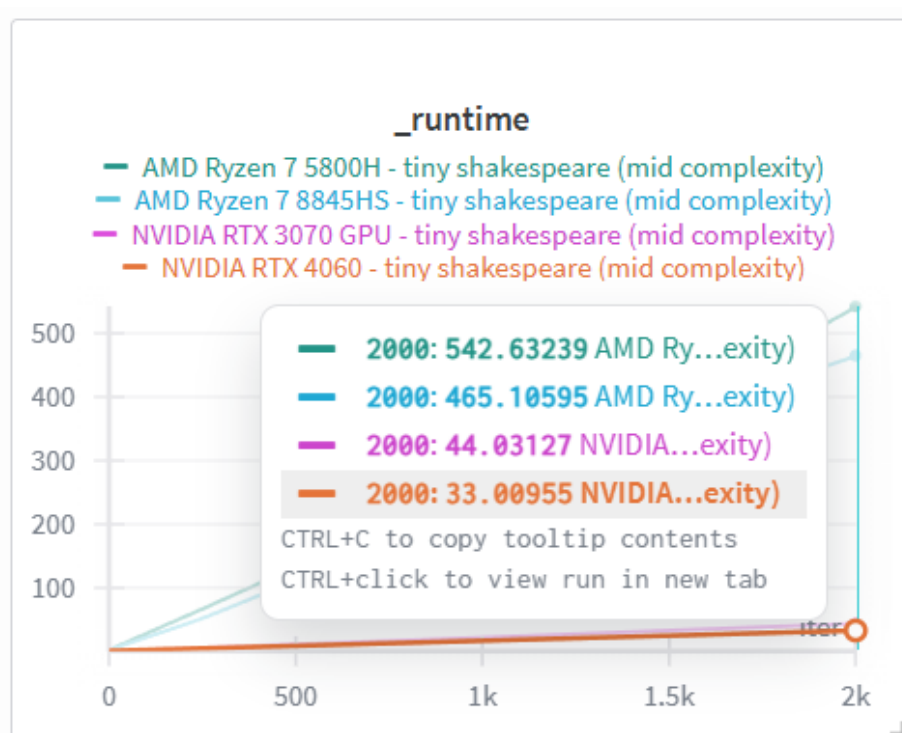
In both machines, **training loss and validation loss curves were nearly identical**. Minor fluctuations were observed between runs as seen in figures 1 & 2; however, these variations were **not correlated with hardware capability**. Even so, when we trained the models on both CPUs as well. What we observed this time around is an abysmal difference in the time it took to train this more complex model. This can be seen in figure 3, all time values are in seconds.



▲ Figure 1



▲ Figure 2



▲ Figure 3

Notably:

- The RTX 4060 occasionally achieved slightly lower loss than the RTX 3070.
- Both AMD Ryzen 7 CPUs performed comparably to both GPUs in terms of train/loss and val/loss but did not do well in total runtime.
- Differences in loss values were minimal and well within expected stochastic noise.
- Compared to the first experiment, we noticed that the val/loss and train/loss both went down by around 0.15 and 0.17 respectively.

At this scale, the model converges smoothly and predictably, and hardware advantages in parallelism did translate in this case into faster outcomes but not into improved value optimization outcomes since both GPUs and CPUs performed similarly in terms of train/loss and val/loss.

These results indicate that for small transformer models with medium depth and moderate learning rates, the optimization landscape is stable enough that different hardware platforms converge to essentially the same solution.

Interpretation

The lack of observable divergence is expected and highlights an important insight:

Hardware affects training speed, not training dynamics, when the optimization problem is simple and numerically stable.

At shallow depth:

- Gradients are well-conditioned
- Floating-point error does not accumulate significantly
- Attention mechanisms are simple and robust
- Optimization follows a deterministic trajectory

As a result, even significantly different hardware platforms produce nearly identical loss curves.

This demonstrates that **high-end hardware is not necessary** for training or experimenting with small character-level language models.

Conclusion

This experiment shows that when training a medium-depth, medium-capacity transformer, differences in hardware specifications (GPU tier vs CPU-only systems) do **not meaningfully impact model convergence or final loss values**. However, parallelism provided by GPUs does have a dramatic effect in output time. At this scale, the optimization problem is sufficiently stable that all systems reach similar solutions, regardless of computational power.

These findings reinforce the idea that **hardware advantages become relevant only when numerical instability, scale, or optimization difficulty are introduced**. For introductory models and small-scale experiments, accessible hardware is sufficient for achieving equivalent learning outcomes but they may take longer to produce these results.